**Objetivo**: Trabajar con estructuras

**La salida del programa debe ser como en el ejemplo.**

**Ejercicio 1.-** Escribir un **programa** que:

* **Lea** un fichero de texto que contiene las coordenadas de una serie de puntos del plano (fichero descrito a continuación), y **cree** una lista, ordenada descendentemente, con los puntos más alejados del origen de coordenadas. Es decir, en la lista, en las posiciones más bajas se encuentran los puntos más alejados del origen.
* **Escriba** la lista obtenida.

El programa deberá utilizar la siguiente **declaración de estructuras** que representan información de un punto del plano (abscisa y ordenada) y de una lista de puntos (ya utilizadas en una práctica anterior):

#define N 20 //maxima cantidad de puntos a considerar

struct Tpunto { // punto en un plano, x abscisas, y ordenadas

float x;

float y;

};

struct TlistaPuntos { //lista de puntos del plano

int tam; //cantidad de valores en el vector puntos

struct Tpunto puntos[N];

};

y las siguientes funciones:

1. De la primera parte: Función *distancia*
2. De la primera parte: Función *leerPunto*:

void leerPunto (FILE \*f, struct Tpunto \*p);

1. De la primera parte: Función *escribirLista*:

void escribirLista (struct TlistaPuntos l);

1. Función *aniadirPunto*:

void aniadirPunto (struct TlistaPuntos \*l, int pos, struct Tpunto p);

Que añade a la lista apuntada por l, el punto p, en la posición pos.

Puede suponerse que hay capacidad en el vector de puntos para una nueva componente y que el valor de pos es correcto (entre 0 y tamaño)

1. Función *buscarPos:*

int buscarPos (struct TlistaPuntos l, struct Tpunto p);

Que, a partir de una lista de puntos, ordenada según la distancia al origen de coordenadas, y un punto, devuelve la posición del primer punto de la lista que se encuentre a distancia menor que el punto p. En caso de no existir tal punto, decide un valor de retorno

1. Función *crearListaOrd:*

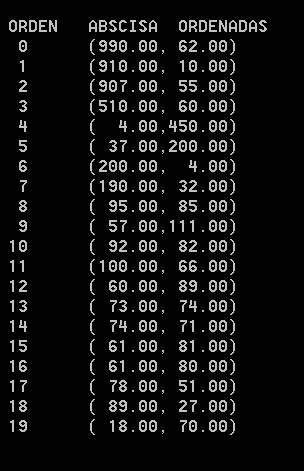
void crearListaOrd (struct TlistaPuntos \*l);

o

struct TlistaPuntos crearListaOrd (void);

Que lee un fichero de texto, variable interna a la función, no vacío, que contiene la abscisa y ordenada de una serie de puntos del plano y almacena en la lista apuntada por l, los N (constante definida) puntos más alejados del origen de coordenadas.

Para el fichero dado, la salida debe ser:



Copia a continuación el programa desarrollado: